

Übung 01: Zahlensysteme

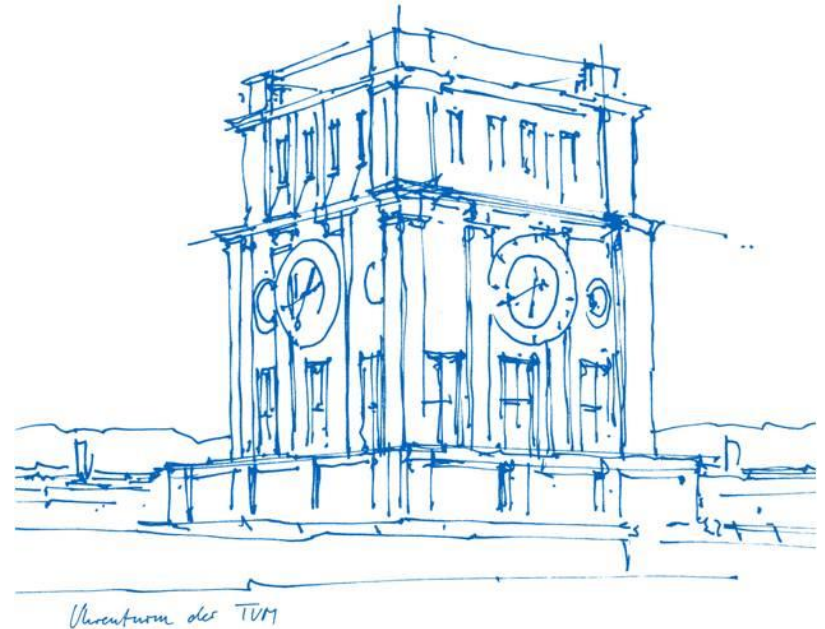
Einführung in die Rechnerarchitektur

Michael Morandell

School of Computation, Information and Technology (CIT)

Technische Universität München

21. Oktober 2024



Disclaimer

Keine Garantie für die Richtigkeit der Tutorfolien.
Bei Unklarheiten/Unstimmigkeiten haben VL/ZÜ-Folien recht!

- Organisatorisches
- Kurze Wiederholung
- Tutorblatt
 - 1) Zahlensysteme
 - 2) Arithmetik und negative Zahlen
 - 3) Zahlenbereiche
- Hausaufgabe

Organisatorisches

Wer bin ich?



- Michael Morandell
- m.morandell@tum.de
- Aus Südtirol (Italien)
- Informatikstudent, 3. Semester
- Meine Tutorien:
 - Mo-1000-4 von 10:00-12:00 im Raum [03.13.010](#)
 - Do-1200-2 von 12:00-14:00 im Raum [01.06.020](#)

Ablauf Tutorium (Mo-1000-4)

- Dauer: ca. 1,5 Stunden
- Beginn um 10:00 oder 10:15?
- Kurze(!) Wiederholung (Konzepte aus der Vorlesung sollten bereits bekannt sein)
- Bearbeitung der Tutorübungen
- Persönliche Fragen



- Mitschriften/Folien auf meiner Homepage: <https://home.in.tum.de/~momi/era/>
- Zulip-Tutorium-Kanal für Mitschriften/Folien/Ankündigungen/:

<https://zulip.in.tum.de/#narrow/stream/2668-ERA-Tutorium---Mo-1000-4>

Ablauf Tutorium (Do-1200-2)

- Dauer: ca. 1,5 Stunden
- Kurze(!) Wiederholung (Konzepte aus der Vorlesung sollten bereits bekannt sein)
- Bearbeitung der Tutorübungen
- Persönliche Fragen



- Mitschriften/Folien auf meiner Homepage: <https://home.in.tum.de/~momi/era/>
- Zulip-Tutorium-Kanal für Mitschriften/Folien/Ankündigungen/
<https://zulip.in.tum.de/#narrow/stream/2657-ERA-Tutorium---Do-1200-2>

Wie bekomme ich Hilfe?

- Am besten in dieser Reihenfolge:
 1. STRG+F Folien durchsuchen
 2. Google
 3. Kommilitoninnen und Kommilitonen
 4. Zulip Suchfunktion nutzen
 5. Zulip: Öffentliche ERA-Channels
 6. Zulip: Tutoriums-Channel
 7. Zulip: DM an Tutor (nicht gerne gesehen!)
 8. Mail an Tutor (nicht gerne gesehen!)


Achtung:


Fragen immer in den richtigen Zulip-Channels stellen und Fragen nicht doppelt stellen (Suchfunktion!)


Nur so können wir garantieren, dass jede Frage beantwortet werden kann

Hausaufgaben

- Registrierung auf Artemis <https://artemis.cit.tum.de>
- Bei 80% der erreichten Punkte: Notenbonus um eine Notenstufe (0.3/0.4) auf bestandene Klausur
- Wöchentliche Aufgabe
- Veröffentlichung: Freitags nach Zentralübung (ZÜ)
- Bearbeitungszeit: 10 Tage
- Achtung: Harte Deadline!

 H01 - Zahlensysteme





You have not started this exercise yet.

Due Date: in 8 days

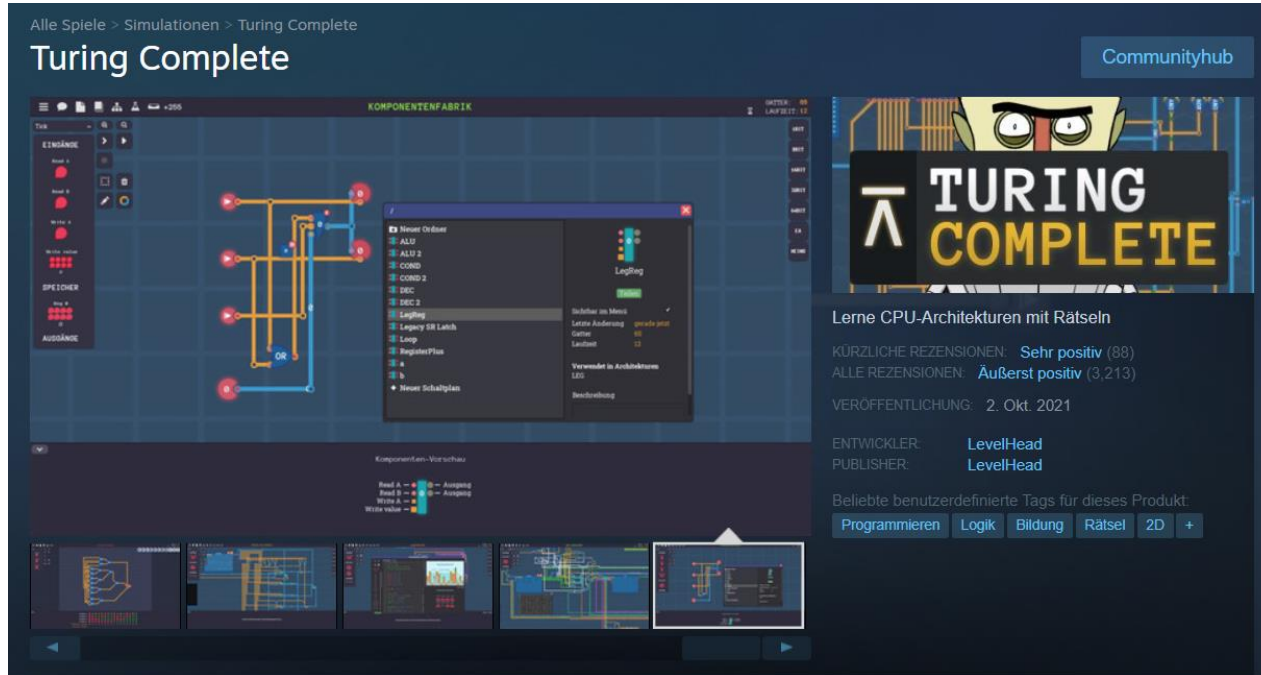
Hausaufgaben

1. Repository-Link auf Artemis kopieren
2. Git-Bash: `git clone <Repo-Link>`
3. Aufgabe bearbeiten
4. In Kommandozeile/Git-Bash: `git add <Dateiname>`
5. In Kommandozeile/Git-Bash: `git commit -m „Commit message“`
6. In Kommandozeile/Git-Bash: `git push`
7. Wiederhole Schritt 3 bis 6 bis Artemis-Tests 100% anzeigen ;)

- <https://rogerdudler.github.io/git-guide/>
- https://learngitbranching.js.org/?locale=de_DE

Kleiner Tipp am Rande

- https://store.steampowered.com/app/1444480/Turing_Complete/



Wiederholung

Was macht so ein Rechner eigentlich?

- Rechner arbeiten mit Zahlen
Bilder, Texte usw. werden mit Zahlen repräsentiert
- Dezimalsystem ungeeignet, da zu viele verschiedene Zustände (10)
- Strom an und aus → Binärsystem (0/1)
- **Designentscheidung!**

Wie funktioniert denn so ein Zahlensystem?

$$W = \sum_{i=0}^{n-1} a_i \cdot B^i \quad (1)$$

Beispiel.

1. $(1010)_2 = 0 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3 = 2 + 8 = 10$
2. $(763)_8 = 3 \cdot 8^0 + 6 \cdot 8^1 + 7 \cdot 8^2 = 3 + 48 + 448 = 499$
3. $(123)_{16} = 3 \cdot 16^0 + 2 \cdot 16^1 + 1 \cdot 16^2 = 3 + 32 + 256 = 291$

$$Z = \sum_{i=0}^n a_i \cdot b^i = a_0 \cdot b^0 + a_1 \cdot b^1 + a_2 \cdot b^2 + \dots + a_n \cdot b^n$$

Einfaches Sign-Bit: (Bsp.: 5)

$$5_{10} = 0000.0101_2 \quad -5_{10} = 1000.0101_2$$

Einerkomplement: (Bsp.: 5) -> Bits flippen

$$5_{10} = 0000.0101_2 \quad -5_{10} = 1111.1010_2$$

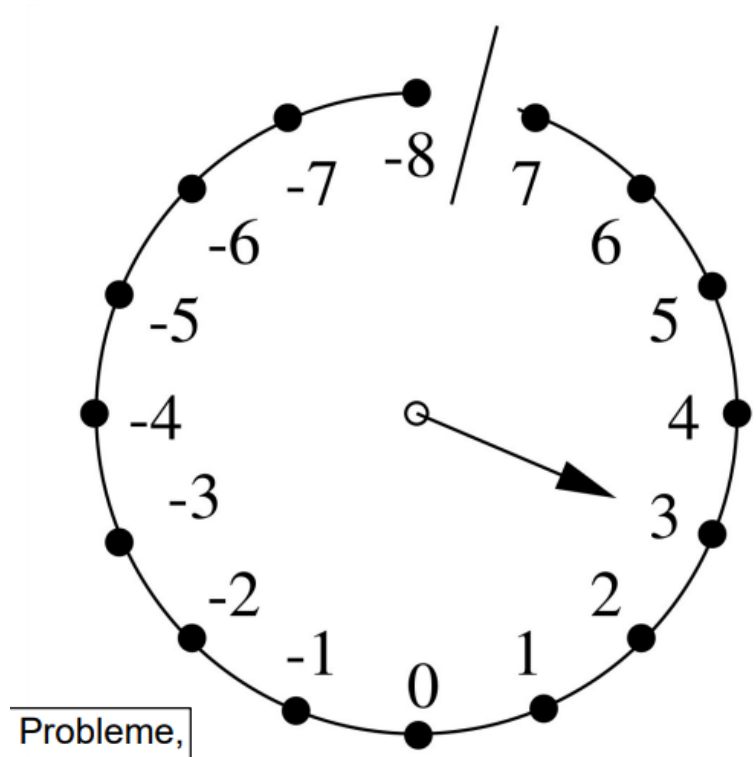
Zweierkomplement: (Bsp.: 5) -> Bits flippen und 1 addieren

$$5_{10} = 0000.0101_2 \quad -5_{10} = 1111.1011_2$$

-> Arithmetik

Wichtig:
Nur wir als **Programmierer / Compiler** wissen, was diese Zahl darstellt, wie sie interpretiert und dementsprechend behandelt werden soll

Overflow



Typname in C	Bits	Vorzeichen	min	max
char	8	ja	-128	127
unsigned char		nein	0	255
short	16	ja	-32.768	32.767
unsigned short		nein	0	65.535
int	32	ja	-2.147.483.648	2.147.483.647
unsigned int		nein	0	4.294.967.295
long long	64	ja	-9.223.372.036.854.775.808	9.223.372.036.854.775.807
unsigned long long		nein	0	18.446.744.073.709.551.615

Tutorblatt

Hausaufgabe

? Teilaufgabe a) No results

Lösen Sie die folgenden Rechenaufgaben (alle Zahlen sind positiv):

- $0001.1011_2 + 0010.1010_2 = (?)_2$
- $0011.0110_2 - 0010.1010_2 = (?)_2$
- $0001.0110_2 * 0000.0101_2 = (?)_2$

? Teilaufgabe b) No results

Berechnen Sie den Wert des Terms $0001.0100_2 - 0100.1110_2 = (?)_2$ indem Sie den Subtrahenden negieren und anschließend auf den Minuenden addieren. Verwenden Sie das **Zweierkomplement** und rechnen Sie mit **8-Bit-Zahlen**. Tragen Sie den negierten Subtrahenden sowie das Ergebnis an den vorgesehenen Stellen in die Datei `submission.txt` ein.

Fragen?

Bis zum nächsten Mal ;)

Diese Folien wurden von Foliensätzen von
Clemens Schwarzmann und Niklas Ladurner inspiriert